International Journal of Computer Networks and Distributed Computing
Vol. 2, Issue 1

A Clustering Method for Peer-to-Peer File Indexing in Wireless Mobile Networks

Barzeen Bin Neor [1], M. Romnek Selemin[2], Sheim Reabulm Iim [3]
Department of Computer Science and Information Technology[1, 2, 3]
Islamic University of Technology (IUT), Gazipur, Bangladesh[1, 2, 3]

ABSTRACT: **Peer-to-peer computing is a popular paradigm for different applications that allow direct message passing among peers. The existing P2P search algorithms in MANET (Mobile Ad-hoc Network) are flooding-based search that produces much traffic and network overhead. File searching efficiency of peer-to-peer (P2P) network mainly depends on the reduction of message overhead. This paper deals with a full form of cluster based P2P file searching approach for MANETs, which focuses mainly over reduction of control messages. For searching, our mechanism uses cluster head within the cluster but it also allows inter-cluster communication. Moreover, secondary cluster head concept of our proposal ensures lower message overhead during cluster formation stage. Our clustering scheme utilizes request suppression to reduce the number of responses for searching process. Cconsideration of alternative paths to a node facilitates to overcome link failure. In detailed simulation, we have showed that our clustering approach for file searching outperforms the existing ORION approach in case of reducing message overhead.**

Keywords: MANET, P2P, Cluster, Cluster Head, RSSI

I. INTRODUCTION

Now-a-days applications based on peer-to-peer (P2P) file sharing are immensely popular among users. Mobile ad hoc networks (MANETs) and P2P networks are both decentralized, exhibit lack of prior infrastructure and hence organized dynamically. Due to above similar aspects, researchers are interested to deploy P2P file sharing over MANETs. But compared to wired Internet this is much more challenging for the reason of wireless restrictions like limited energy and processing power, low bandwidth, unreliable physical channel, less memory as well as node mobility[1][2]. Moreover, MANETs have to work with the dynamic topology at both application and physical level whereas wired network have to deal with the dynamic topology only at application level [3]. These specifications create difficulties to implement P2P file sharing in MANET compared to traditional wired P2P sharing.

For developing a P2P file sharing over MANET, reducing message overhead is a major issue that ensures power efficiency and increased scalability of the system [4]. The amount of signaling message overhead increases with the growth of network size. Moreover, maintaining scalability and stability of dynamic topology also need a large amount of message transfer among nodes in MANET. To reduce signaling overhead, clustering is a popular technique in MANETs that organize nodes (hosts) in a network into logically separated units called clusters [5]. Each cluster consists of a cluster head (CH) node and some ordinary member nodes. Cluster head selection is an important factor as it impacts on intra-cluster communication as well as on inter–cluster communication. Furthermore, in order to reduce overhead, number of cluster must be minimized and also the stability of the clusters has to be increased. Peer-to-peer (P2P) file sharing application needs both file searching and file transfer mechanism to complete the total process. During the searching phase, minimizing the number of messages is again a significant issue [6].

In this paper we focus on the single hop clustering approach from cluster head. A cluster is formed based on the cluster head's transmission range, mobility and power. In addition, a method has been proposed with maintaining secondary cluster head that increases the stability of the cluster and the efficiency of overall searching process. During the searching process from any destination, alternate paths to the desired file have been identified in order to overcome link failure while file transfer. Finally we have compared our proposed clustering scheme for P2P file searching with ORION [7] approach in respect of message overhead.

The remainder of this paper is structured as follows: The related work on clustering approach and P2P sharing in MANET is briefly described in Section II. Section III consists of our proposed mechanism. Results are presented and analyzed in Section IV. Finally, a summary of our conclusion and future direction are given in Section V.

II. Related Work

Some protocols like ORION [7], MPP [8] [9], BTM [10] were specially designed to address MANETs limitations in peer-to-peer file sharing. In Dynamic downloading mode of MPP approach, only the first path acquired by the file search stage is used for file transfer which results in a new search initiation in case of link failure. BTM and ORION protocol requires each node to maintain overlay connections that cause considerable overhead for a MANET with a large number of nodes. In reactive path selection technique, ORION and MPP (Static downloading mode) cache the paths acquired by the file search at the requesting node. In response to path failure and to avoid frequent flooding during file searching, ORION+ [11] was proposed where new unicast messaging mechanism to the reactive flooding technique was applied to ensure successful file transfer in a minimal time. Deluge [12], an another approach of P2P file sharing uses the concept of network coding where multipoint communication reduces energy consumption and file transfer time. However, the searching phase and initial processing of searching result of a request is not well discussed in Deluge.

A cluster-based P2P file sharing protocol for MANET [13] mainly focused on reducing message overhead during file transfer process. This protocol applies three way handshaking to form a cluster staring with a node interested in file sharing. Though the clustering concept of this protocol improves file transfer, more emphasize would be given on efficient clustering of nodes. A number of clustering approach [14] [15] has been proposed so far and these deals with full form cluster formation with cluster head selection. DEECA [16] is another energy efficient clustering approach for MANETs where initially node with the highest energy level and less mobility is elected as cluster head. Chord [17], a well-known decentralized P2P search algorithms, improves the scalability by avoiding the requirement that every node knows about each other node [18] which is different from flooding-based search algorithm such as Gnutella.

III. Proposed Mechanism

Our proposed approach forms a single-hop cluster from its Cluster-Head (CH) to member node. After the formation of cluster, protocol is shifted to phase of file searching in peer-to-peer manner. Inter-Cluster communication occurs when the desired node resides outside of cluster from which the searching query initiates. The stability of cluster is also maintained throughout the process. The details of our method consisting of new node initialization, cluster formation, cluster maintenance and file searching phase is described below.

A. Initialization of a Node

When a node, say X, is powered up, it sets its state value as 0, indicating its initialization phase. It sends HELLO message to all of its neighbouring nodes with the information (node ID, energy level, CH ID) of initializing node. Default CH ID of initializing node's HELLO message is set to 0. Upon receiving the HELLO message, following neighbouring nodes reply node X with HELLO message:

- ✓ Cluster-Head itself when the initializing node is within the range of existing cluster
- ✓ The node that is in initializing phase which represents the node still is not a member of a cluster and so having no CH information

By comparing the node ID and CH ID of received HELLO message, initializing node determines whether the reply came from CH or from another initializing node. Same node ID and CH ID indicate that the replied HELLO message is from cluster head.

However, the initializing node can receive HELLO message either from single CH or from multiple CHs. Upon receiving message from single CH, initializing node X simply joins to the cluster by sending JOIN message to CH and CH considers the node as one of his cluster member as Fig. 1. JOIN message contains the information of node ID, energy level, filename and data block range of the file. Here we are assuming that a file is a collection of data blocks with predetermined size. So when CH receives a JOIN message with the value (X, 50, A, 0-12) it considers that a new node (X) has joined this cluster with energy level 50 and the node contains file 'A' with data block range 0-12. Upon receiving reply from multiple CHs, the initializing node X checks energy level of those CHs and sends JOIN message to the higher one.

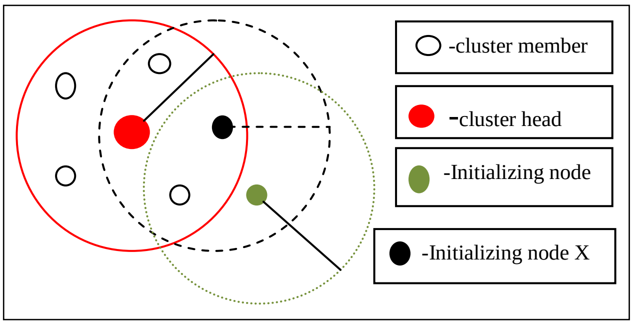


Fig. 1 Initializing node X receives message from single cluster head

On the other hand, when the node X does not receive any reply message from a CH, it realizes that there exists no CH within its transmission range. In this phase the initialling node X enters into cluster formation stage.

B. Cluster Formation and Cluster-Head Selection

The cluster formation phase is invoked when initializing node X does not receive any reply message. Cluster is formed primarily for the following two circumstances:

- ✓ Though there remain some member nodes of a cluster but unfortunately CH of that cluster is out of the range for initializing node X.
- ✓ The node X has no neighboring node(s) within its transmission range.

For these cases, cluster is formed with node X declaring itself as a Cluster Head as Fig. 2. This newly formed cluster does not contain any cluster member because CH has no neighboring nodes for second case and for first case, though there remain some neighboring nodes of X, they are already a member of another cluster.

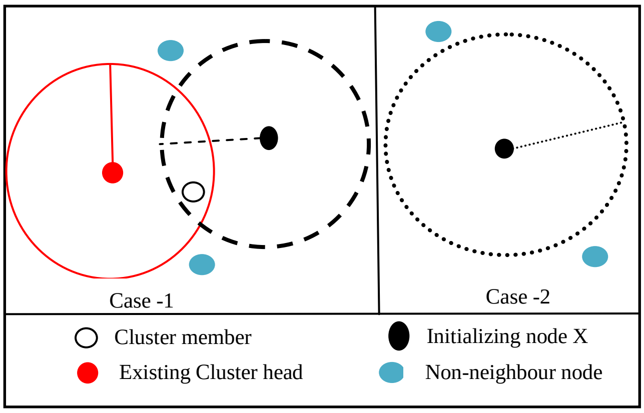


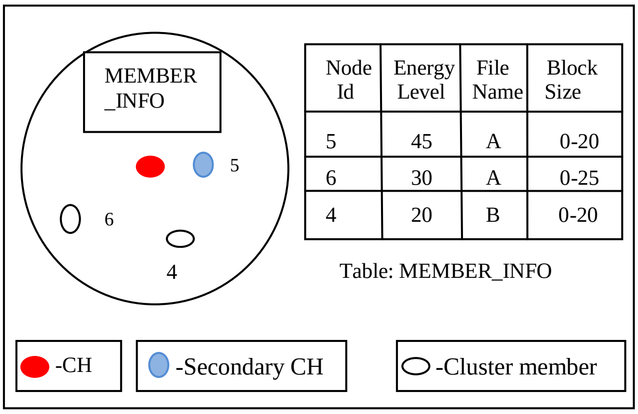
Fig. 2 Initializing node X receives no reply message

On the other hand, if node X receives reply message from neighbouring nodes but none of these node is CH then X considers that the neighbouring nodes are still in initialization phase. These nodes individually form clusters but the question arises regarding the selection of Cluster-Head.

Node X compares energy level of each neighbouring node with itself. In this case, when node X finds that its energy-level is less than any of the energy-level of its neighbouring nodes, then node X expects that the node (Y) having the higher energy level will form a cluster by assigning itself (Y) as a cluster head. In the mean time node Y forms a cluster assigning itself as CH and sends message to its neighbouring nodes about its CH state. After receiving this message, node X joins to the cluster that is recently formed by node Y and also informs other nodes in initializing state having lower level energy than X. Upon receiving this message the nodes with lower energy level again send HELLO message and the same process repeats until a node becomes CH or the node becomes a member of another cluster. The node joins to a new cluster by sending JOIN message to the CH of that cluster. Based on the JOIN message CH adds the new node as a member in the MEMBER_INFO table of CH node.

C. Secondary Cluster Head Selection

Our proposed method consists of selecting a Secondary Cluster Head among the members of the cluster. It will become primary CH when the previous CH suddenly leaves the cluster or the energy-level become less than a threshold value. This approach increases stability of the cluster by immediately transferring control of primary CH to the secondary cluster. The Primary CH selects the Secondary one by comparing the energy level of all its member nodes from MEMBER_INFO table and the node with the highest energy-level is selected as the Secondary CH. Finally this node Id is broadcasted to all of the member nodes within that cluster. As in Fig. 3, node 5 becomes secondary CH by comparing its energy level with node 6 and 4. The process of maintaining a secondary CH reduces message overhead of new CH selection in cluster formation stage.



Node Id	Energy Level	File Name	Block Size
5	45	A	0-20
6	30	A	0-25
4	20	B	0-20

Fig. 3 A cluster with Secondary CH and MEMBER_INFO table of CH

D. Cluster Maintenance

Cluster-Head sends ALIVE message with its node ID and energy level to the member nodes of the cluster periodically. Let us consider this periodic time is T sec. Before sending the ALIVE message CH starts a timer TA. All the neighbouring nodes of cluster head receive the message and send acknowledgement to the CH. As the timer expired, CH checks its MEMBER_INFO table in order to find the members who have not send acknowledgement message but their entries reside in the table. The information of those nodes is deleted from the MEMBER_INFO table as the nodes are no longer a neighbouring node of the previous CH. On the other hand, all the member nodes except Secondary CH wait for T period. If it does not receive any ALIVE message from its CH, it again waits for T period to get ALIVE message from Secondary CH. When a member node receives ALIVE message from Secondary CH, it joins under that CH by sending JOIN message. Otherwise the node checks whether it has received ALIVE message from other CHs within this 2T period. If the node receives ALIVE message from other CH, then the node

joins that CH otherwise enters into initializing phase. But the Secondary CH waits T period for receiving ALIVE message from its CH. If it does not receive this, it sends ALIVE message with its node ID and waits T period to receive JOIN message from its neighbour node(s). As in Fig. 4, due to mobility of CH the cluster became obsolete and hence secondary CH became the primary one. This also results in some nodes to enter into initializing phase but obviously the number is less than all member nodes of previous cluster.

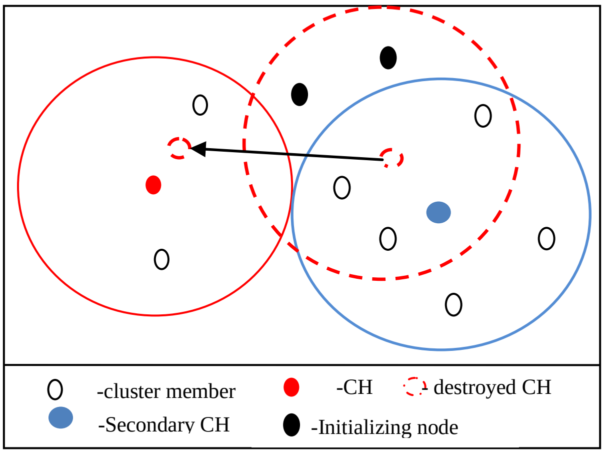


Fig. 4 Cluster maintenance

E. File Searching

When a node initializes search for a file it sends a message FIND with its node id, requesting file name, data block of the file to the CH of its cluster. The CH searches its own MEMBER_INFO table for the requested file. If it finds the file within its members, it sends ID of the node that contains the requested file to the requesting node. After this, requesting node goes for transferring the file from that node via CH. When there remain multiple nodes with the same requested file, CH keeps track of that those alternate sources in its PATH table (TABLE 1). So in case of link failure the alternate path can be used to search the file or to transfer the remaining blocks of data. If the CH cannot find the requested file in its own cluster then it goes for inter cluster communication. For this process, the CH broadcasts the FIND message on behalf of the requesting node to all of its member nodes. Upon receiving FIND messages from CH, member nodes of the cluster search for neighbouring nodes within their transmission range. The node checks out whether any of its neighbouring nodes is a member of different cluster and if so, then the FIND message is propagated to CH of another cluster through the neighbouring node. When there reside no CH of different cluster within the transmission range of member node, the same process of broadcasting FIND message continues. For processing each query request within the cluster, CH initiates cluster update by sending and receiving HELLO to observe current status of its initial member nodes.

TABLE 1
PAH TABLE

Requesting Node ID	File name	Requesting data block	Source Node ID	Next Node ID
1	A	0-5	2	5
1	A	0-5	3	5
1	A	0-5	3	4

Communication with other cluster differs in various circumstances. When there remains a CH within the transmission range of a member node, it directly sends the message to the CH as case-1 in Fig. 5. If multiple paths are available to communicate with another cluster then optimal path is chosen based on strong RSSI signal [19] [20] to broadcast the message as case-2 of Fig. 5. The other nodes overhear the message but the alternative paths are stored in the PATH table for secure communication. On the other hand, when there remain multiple nodes to communicate with the same member of another cluster as case-3 of Fig. 5, then each of the nodes broadcast message to that single member of other cluster. The member then chooses the optimal one by measuring RSSI signal of all sending nodes and stores the information into its PATH table along with other alternatives.

In case of receiving reply from different destinations for the same requested file block, optimal one is chosen based on the hop count and the round trip time of earlier messages. Reply messages follow the path that was stored in PATH table during search phase, but in a reverse way. Besides, it is also possible that the requested file is shared among different nodes. Optimization is done here by adopting ORION approach where partial file blocks are locally cached into intermediate nodes for faster access in future.

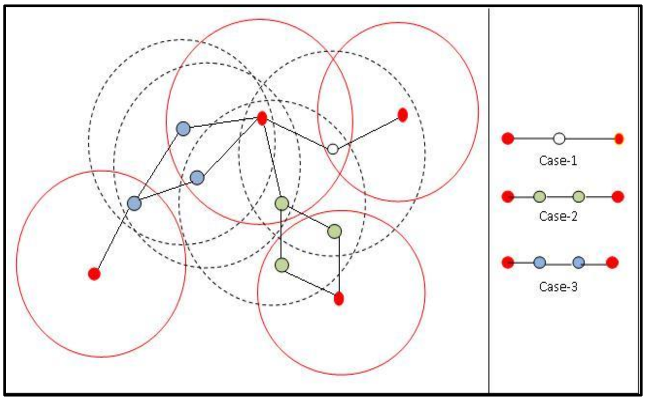


Fig. 5 File searching scenario

IV. PERFORMANCE RESULTS

For the simulation of our proposed mechanism we have used Java platform. The detail procedure of simulation and performance comparison is described below.

A. Simulation Environment

We assume that maximum 100 nodes move in a network area of 1000m X 1000m for our simulation. Transmission range considered of each node is 250 m and mobility model applied is linear mobility at constant speed. Each node in the network is initialized with the parameters: Node ID, Energy, X-coordinate, Y-coordinate, No. of files, File names, File block range. For simplicity, Node ID is initialized with a unique number between 1 and 500. Energy value is randomly set within the value of 100 (%). Initially, X and Y coordinate of each node is randomly set within the network area. We assume each node can have maximum 5 files where each File Name is initiated with single Capital letter (A, B, C….Z). We are also considering that each file can have maximum 10 file blocks. For our proposed clustering approach we set the initial value of Cluster-Head and Secondary Cluster-Head is 0.

B. Clustering Performance

For simulation of clustering approach, initially we form clusters with 10 nodes based on their energy value. For this phase we apply both the existing approach and ours one on the same network maintaining same cluster structure. This simulation is run by 10 times and the average number of cluster is taken. We have considered different scenario as proposed and measured performance based on these cases. This procedure is repeated after increasing the number of nodes in the network by 10. Fig. 6 shows the corresponding graph of number of cluster for different number of nodes in the network.

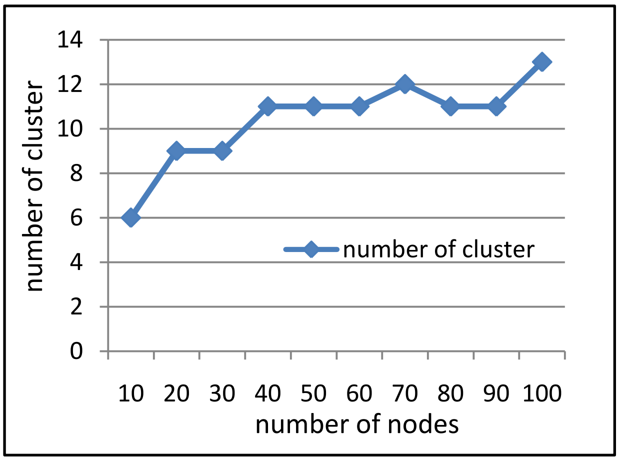


Fig. 6 Number of cluster over nodes

Member nodes of a cluster become leaderless when cluster head (CH) of that cluster moves away due to node mobility or its energy level become less than threshold value. The nodes with no CH enter into initialization phase. Fig. 7 illustrates the average number of nodes enter into initialization phase for both using primary CH and secondary CH approach when Cluster-heads randomly become unavailable. Here the average value of 10 simulations has been taken into consideration. The fluctuations occur in both approaches as random number of CH become unavailable. However, the graph clearly shows that using secondary cluster-head provides less number of nodes entering into initializing phase.

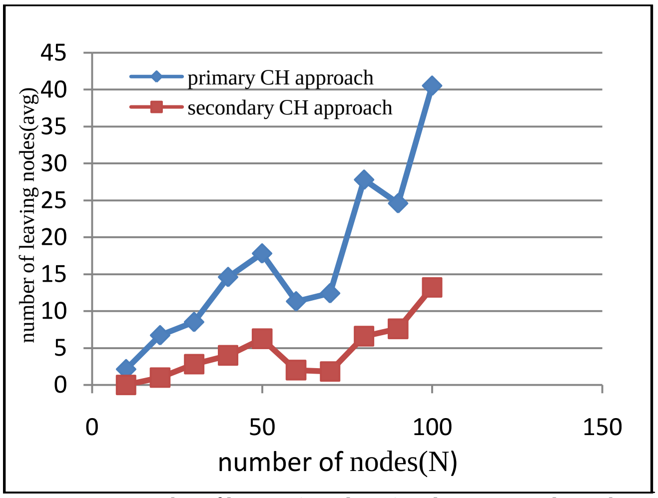


Fig. 7 Average number of leaving (initializing) nodes over N nodes in the network

When these nodes return in initialization phase, they need to transfer messages for cluster reformation. The more number of nodes enter into the phase the more number of messages is transferred for cluster reformation.

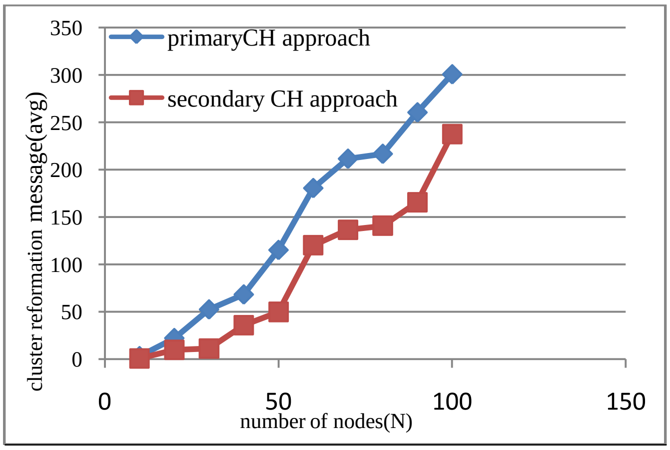


Fig. 8 Average cluster reformation message over N nodes in network

Fig. 8 compares new message overhead for cluster reformation between the approaches using primary CH and secondary CH. As an overall trend it is acknowledged that clustering with secondary CH generates fewer messages overhead in the network.

C. Searching Performance

We compare our searching approach with ORION [7] technique. At first we have implemented ORION in our network. In this approach, a node requests for a number of blocks of a file. This request is broadcast to its neighbour node. If the neighbour node contains the complete blocks of request then it replies to requesting node. However, when the neighbouring node contains partial blocks of a file, it again broadcast the modified search request in order to get unavailable file block from other nodes.

In our clustered approach, we are applying the ORION concept with some modifications. Our mechanism deals with clusters having Cluster-Head that contains information regarding the member nodes of the cluster. Here at first search query requests are sent to cluster-head. Cluster-head checks its information table to identify the member nodes having the requested file name with file block and reply accordingly. If the query is not fulfilled within the cluster then CH broadcast request message to other cluster via neighbouring nodes. Each neighbouring node communicate with other cluster head either directly (if CH is within its range) or via another member node. Throughout this process if any node contains requesting block, it replies to the source via intermediate nodes and the intermediate nodes keep the information of the file blocks like ORION approach. As intermediate nodes locally store the file information, so for any further request this approach initiates fewer messages to broadcast.

For performance analysis, we consider 55 nodes in the network. Initially we take 5 search requests for both ORION and for our approach but the number of request are increased by 5 up to 35. After simulating 10 times, the average number of transferred query message to process the searching requests for both of the approaches has been plotted in Fig. 9.

As Fig. 9 indicates, using clustered approach for file searching reduces the number of query messages than ORION approach. In clustering approach, as CH contains information of its member nodes, so it replies the information to requesting node without broadcasting the message. In ORION the search request query is frequently broadcast to all nodes in the network. For this reason the number of query message in clustered approach is less than the ORION technique.

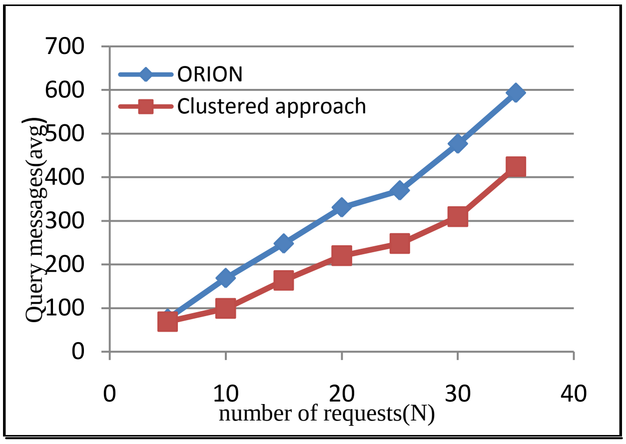


Fig. 9 Average number of query message for N requests

V. Conclusion

In this paper, a well organized P2P file searching in MANET has been proposed. A set of constraints has been considered that makes the searching process more convenient. Our mechanism shares local storage feature of ORION within the nodes but as this is applied in a clustering method, it provides lower message overhead. Moreover, secondary cluster head maintenance also reduces control messages during cluster formation. In future, we will verify our proposed approach using probabilistic analysis.

References

[1] Y.C. Hu, S.M.Das and H. Pucha, “Peer-to-Peer Overlay Abstractions in MANETs,” Book Chapter, in *Theoretical and Algorithmic Aspects of Sensor, Ad Hoc Wireless, and Peer-to-Peer Networks*, Jie Wu, editor, CRC Press, 2005.

[2] A. Hayes and D. Wilson, “Peer-to-Peer Information Sharing in a Mobile Ad Hoc Environment”, *in Proceedings of Sixth IEEE Workshop on Mobile Computing Systems and Applications (WMCSA'04)*, 2004, p. 154-162.

[3] J.Eberspächer, R. Schollmeier, S. Zöls, and G.Kunzmann, ”Structured P2P Networks in Mobile and Fixed Environments*”, In proceedings of HETNETs '04*, July 2004.

[4] Rim Haw, Choong Seon Hong, and Dae Sun Kim, “Group P2P Network Organization in Mobile Ad-hoc Network,” *in Proc. APNOMS'09*, 2000, p. 475-480.

[5] S. Chinara and S. K. Rath, “A survey on one-hop clustering algorithms in mobile ad hoc networks,” *Journal of Network and Systems Management*, vol. 17, no. 1-2, pp. 183–207, 2009.

[6] (2012) Gnutella Protocol Development. [Online]. Available: http://rfc-gnutella.sourceforge.net/src/rfc-0_6-draft.html

[7] A. Klemm, C. Lindemann, and O. Waldhorst, A Special-Purpose Peer-to-Peer File Sharing System for Mobile Ad Hoc Networks, in *Proc. IEEE Semiannual Vehicular Technology Conference (VTC-FALL 2003)*, 2758-2763, Orlando, FL, USA, 2003.

[8] A R. Schollmeier, I. Gruber, and F. Niethammer, “Protocol for Peer-to-Peer Networking in Mobile Environments”, *In Proc. of 12th IEEE International Conference on Computer Communications and Networks*, USA, October 2003.

[9] I. Gruber, R. Schollmeier and W. Kellerer, “Performance Evaluation of the Mobile Peer-to-Peer Protocol”, *In Proceedings of Fourth International Workshop on Global and Peer-to-Peer Computing,* Chicago, USA, 2004.

[10] Sundaram Rajagopalan, Chien-Chung Shen, “A Cross-layer Decentralized BitTorrent for Mobile Ad hoc Networks,” *Third Annual International Conference on Mobile and Ubiquitous Systems: Networking & Services*, San Jose, CA, July 2006, p. 1-10.

[11] A. Abada, L. Cui, C. Huang, and H. Chen, "A novel path selection and recovery mechanism for MANETs P2P file sharing applications," *in Proc. WCNC*, 2007, pp.3472-3477.

[12] Afzal Mawji and Hossam Hassanein,"Efficient Multipoint P2P File Sharing in MANETs", *in Proceeding GLOBECOM'09,* 2009, P. 1253-1258.

[13] Junmin Jia,Chun Meng,"A Cluster-based Peer-to-Peer File Sharing Protocol for Mobile Ad Hoc Networks", *International Symposium on CNMT 2009, p.1-4.*

[14] Hwa-Chun Lin ,Yung-Hua Chu. “A Clustering Technique for Large Multihop Mobile Wireless Networks". *Vehicular Technology Conference Proceedings*, 2000. Vol. 2, p. 1545 – 1549.

[15] Yi, Y., Gerla, M., Kwon, T. J., “Efficient Flooding in Ad hoc Networks using On-Demand (Passive) Cluster Formation”, *ACM Symposium on Mobile Ad Hoc Networking* **,** MOBIHOC 2002.

[16] Haidar Safa,Omar Mirza, Hassan Artail,"A Dynamic Energy Efficient Clustering Algorithm for MANETs”, IEEE International Conference on WIMOB '08,2008, p. 51-56.

[17] (2011) Chord project. Available: http://pdos.lcs.mit.edu/chord/

[18] Ion Stoica, Robert Morris, David Karger, M. Frans Kaashoek, Hari Balakrishnany:‘Chord: A Scalable Peer-to-peer Lookup Service for Internet Applications”, *IEEE/ACM Transactions on Networking*, vol. 11, issue 1, pp. 17-32.

[19] Rong-Hou Wu, Yang-Han Lee, Hsien-Wei Tseng, Yih-Guang Jan and Ming-Hsueh Chuang,“Study of characteristics of RSSI signal”, *IEEE International Conference on Industrial Technology*, ICIT 2008, p. 1-3.

[20] *Wireless Medium Access Control (MAC) and Physical Layer (PHY) Specifications for Low-Rate Wireless Personal Area Networks*, IEEE Std. 802.15.4, 2006.

Biography

Tanzeem Bin Noor is currently active as lecturer in Department of Computer Science and Engineering (CSE), Stamford University Bangladesh. He received his BSc degree in Computer Science and Information Technology from Islamic University of Technology (IUT), Gazipur, Bangladesh in 2010. His research interest is mainly focused on Parallel and Distributed System, Ad Hoc Networks, Software Methodology, Bioinformatics and Image Processing.

Md. Rounok Salehin obtained his BSc degree in Computer Science and Information Technology from Islamic University of Technology (IUT), Gazipur, Bangladesh in 2010. He worked as a lecturer in Department of Computer Science and Engineering (CSE), Stamford University Bangladesh. Currently, he is undergoing his MSc degree in Department of Computer Science, University of Western Ontario, Canada. His research topics cover Software Security, Software Requirement Engineering, Ad Hoc Networks, Distributed Computing and Bioinformatics.

Sheikh Rabiul Islam is currently working as an assistant manager (IT and Billing) at Teletalk Bangladesh Ltd.. He received his BSc degree in Computer Science and Information Technology from Islamic University of Technology (IUT), Gazipur, Bangladesh in 2010. His research interests include Ad Hoc Networks, Network Security and HCI.